WIX1002
Fundamentals of Programming

Chapter 3

Flow of Control (Selection)

Contents

- Introduction
- Relational Operator
- Logical Operator
- if
- if-else
- ultiway if-else
- switch
- Ternary Operator
- Common Error

Introduction

- Flow control is to used to specify the order of the statements to be executed.
- Program can be written in three types of flow control namely the **sequence, selection and repetition**.
- When the statements are executed **one after the other** in order, it is called the **sequence flow**.
- A **selection flow** chooses among **alternative courses of action**.
- A **repetition flow** specifies that an action is to be **repeated while some condition remains true**.

Introduction

- Computer programs often need to make decisions, taking different actions depending on the condition.
- In Java, **if** and **switch** statement are used to carry out the decision.
- The statement is controlled by the **boolean expression**.
- **Relational/Conditional operator** can be used in the boolean expression.
- If there is more than one **constraint/condition** in the decision making, the **logical operators** are used to merge multiple constraint/conditions.

Relational Operator

- The relational operator tests the relationship between two values.

Operator	Description	Examples
==	Equal	a==b
!=	Not Equal	c!=d
>	Greater than	x>y
>=	Greater than or equal	x>=y
<	Less than	a<b
<=	Less than or equal	a<=b

Logical Operator

- The logical operator is used to create complex Boolean expression by merging multiple constraints/conditions.

Operator	Description	Examples
&&	AND (true && true is true, others false)	a==b && c==d
||	OR (false && false is false, others true)	c!=d || a<b
!	NOT (!false is true. !true is false)	!(x>y)

if

- if statement is used to implement a decision. It consists of condition and body.
- If the condition is true, the body of the statement is executed.

```
if ( condition1 )
    statement 1;

// use brace {                         more than 1 statements
if ( condition1 ) {
    statement 1;
    statement 2;
}
```

if

```
if (number > 0)
    System.out.println("The number is positive");

if (result < 50) {
    System.out.println("You did not pass");
    System.out.println("Try harder next time");
}
```

if-else

- if-else statement chooses between two alternative statements based on the condition or boolean expression

```
if ( condition1 )
    statement 1;
else
    statement 2;

if (myScore > yourScore)
    System.out.println("I Win!");
else
    System.out.println("You Win!");
```

if-else

```
// use brace {
if ( condition1 ) {
    statement 1;
    statement 2;
}
else {
    statement 3;
    statement 4;
}
```

more than 1 statements

if-else

- String Comparison
  - When testing string for equality, **DO NOT USE ==** operator. Use **equals** or **equalsIgnoreCase**.
  - String.equals(other_string)
  - String.equalsIgnoreCase(other_string)

```
String s1, s2;
if (s1.equals(s2))
   System.out.println("They are equal strings.");
else
   System.out.println("They are not equal strings.");
```

if-else

- **Alphabetical Order**
  - Lexicographic ordering is used to order alphabet according to ASCII ordering. Use **compareTo** and **compareToIgnoreCase**.
  - String.compareTo (other_string)
  - String.compareToIgnoreCase(other_string)

  - s1.compareTo(s2)
    - Return **negative** value if **s1 comes before s2**.
    - Return **positive** value if **s2 comes before s1**.
    - Return zero if s1 is equal to s2.

Multiway if-else

- Multiway if-else statement is the if-else statement nested inside the if-else statement

```
if ( condition1 )
    statement 1;
else if ( condition2 )
    statement 2;
else if ( condition3 )
    statement 3;
else
    statement 4;
```

Multiway if-else

```
if (myScore > yourScore) {
    System.out.println("I Win!");
}
else if (myScore < yourScore) {
    System.out.println("You Win!");
}
else {
    System.out.println("Tie!");
}
```

switch

- switch statement can be used to represent multiway if-else statement.

```
switch ( variable ) {
case value1:
    statement 1;
    break;
case value2:
    statement 2;
    break;
default:
    statement 3;
}
```

switch

```
switch (number) {
   case 1:
      System.out.println("Satu");
      break;
   case 2:
      System.out.println("Dua");
      break;
   case 3:
      System.out.println("Tiga");
      break;
   default:
      System.out.println("This program accepts the number from 1 to 3 only");
}
```

Ternary Operator

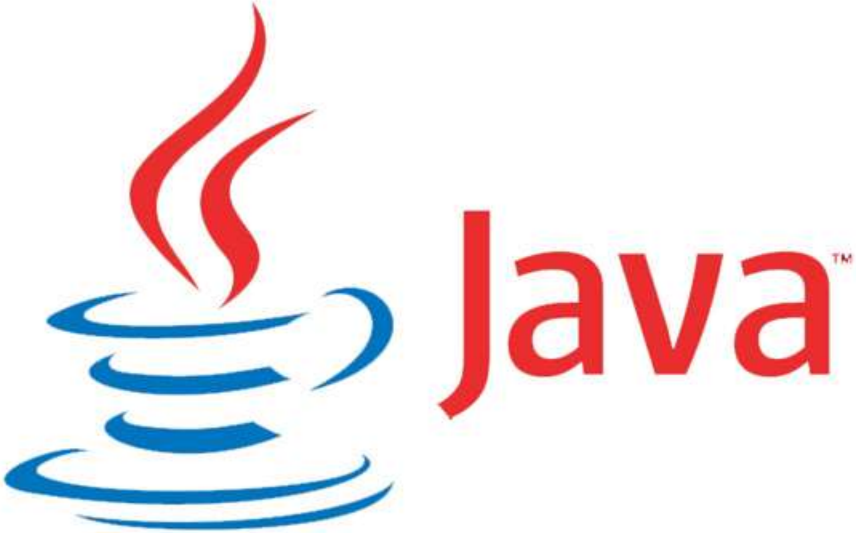
- The ternary operator ? : is similar to if-else statement
- **condition1 ? statement1 : statement2**
  - If the condition1 is true, the statement1 will be executed.
  - If the condition1 is false, the statement2 will be executed.
  - y = x >=0 ? x : -x;

Common Error

- if number > 0
  - System.out.println("No bracket");
- if (x > 5 * (y - z)
  - System.out.println("Miss one bracket");
- if (0 <= mark <= 100)
  - System.out.println("Invalid syntax");
- if (choice **=** 'Q')
  - System.out.println("Equal sign error");
- if (age > 21 and mark > 80)
  - System.out.println("and should be &&");

Common Error

-
  ```
  switch (monthNum) {
    case 1:
        System.out.println("January");
    case 2:
       System.out.println("No break");
  }
  ```


Java™